CLAIMS

Cancel claims 1-30.

Please enter new claims 31-54 as follows:

--31.(New) A method of processing Java format calls, said method comprising the steps of:

at compile time, compiling a library which includes for each of a multiplicity of COM methods, a return value in a COM format for said each COM method, a name of a type of COM object to contain said each COM method and an identifier for said each COM method;

at compile time, compiling from said library a Java interface which includes for each of said COM methods, a return value in a Java format for said each COM method and a name for said each COM method;

at run time, in response to respective requests in Java format from an application to create respective proxy objects for corresponding COM objects, a run time program module generating said respective proxy objects, each of said proxy objects performing the following steps for the corresponding COM object:

receiving a respective call in Java format from the application for the corresponding COM object, and in response, querying said library for the identifier for said corresponding COM method;

transforming said respective call in Java format to a call in COM format, and invoking said corresponding COM method with said call in COM format; and

returning to said application said return value in said Java format.

32.(New) A method as set forth in claim 31 wherein:

said library which is compiled at compile time also includes in COM format for each of said COM methods, types of arguments for a call to said each COM method; and

said Java interface which is complied at compile time, also includes in Java format for each of said COM methods, types of arguments for the call to said each COM method.

33.(New) A method as set forth in claim 32 wherein each of said calls in Java format from an application includes data defining arguments; and further comprising the step of:

at compile time, a Java compiler using said Java interface to check validity of types of said arguments in said each of said calls in said Java format.

34.(New) A method as set forth in claim 32 wherein said identifier for said each COM method, said return value for said each COM method, said types of arguments for a call to said each COM Method, and said COM object containing said each COM method form part of all of a signature for said each COM Method.

35.(New) A method as set forth in claim 32 wherein:

in response to said respective call in Java format from the application for the corresponding COM object, further comprising the step of identifying from said respective call, data defining arguments of said respective call; and

the call transformed into the format of said corresponding COM method includes data defining arguments corresponding to the data defining arguments of said call in Java format.

36.(New) A method as set forth in claim 31 wherein there is a one-to-one correspondence between types of said calls in Java format and corresponding COM methods.

37.(New) A method as set forth in claim 31 wherein before the step of returning to said application said return value in said Java format, further comprising the step of transforming a format of a return value from said corresponding COM object from the format of said corresponding COM object to said Java format.

38.(New) A method as set forth in claim 31 wherein said library which is compiled at compile time also includes a name of said each COM method.

39.(New) A system for processing Java format calls, said system comprising:

means, at compile time, for compiling a library which includes for each of a multiplicity of COM methods, a return value in a COM format for said each COM method, a name of a type of COM object to contain said each COM method and an identifier for said each COM method;

means, at compile time, for compiling from said library a Java interface which includes for each of said COM methods, a return value in a Java format for said each COM method and a name for said each COM method;

means, at run time, responsive to respective requests in Java format from an application to create respective proxy objects for corresponding COM objects, for generating said respective proxy objects, each of said proxy objects including computer programming for the corresponding COM object to:

receive a respective call in Java format from the application for the corresponding COM object, and in response, query said library for the identifier for said corresponding COM method;

transform said respective call in Java format to a call in COM format and invoke said corresponding COM method with said call in COM format; and

return to said application said return value in said Java format.

40.(New)A system as set forth in claim 39 wherein:

said library which is compiled at compile time also includes in COM format for each of said COM methods, types of arguments for a call to said each COM method; and

said Java interface which is complied at compile time, also includes in Java format for each of said COM methods, types of arguments for the call to said each COM method

41.(New) A system as set forth in claim 40 wherein each of said calls in Java format from an application includes data defining arguments; and further comprising:

a Java compiler, at compile time, which uses said Java interface to check validity of types of said arguments in said each of said calls in said Java format.

42.(New) A system as set forth in claim 40 wherein said identifier for said each COM method, said return value for said each COM method, said types of arguments for a call to said each COM Method, and said COM object containing said each COM method form part or all of a signature for said each COM Method.

43.(New) A system as set forth in claim 40 further comprising:

means, responsive to said respective call in Java format from the application for the corresponding COM object, for identifying from said respective call, data defining arguments of said respective call; and wherein

the call transformed into the format of said corresponding COM method includes data defining arguments corresponding to the data defining arguments of said call in Java format.

44.(New) A system as set forth in claim 39 wherein there is a one-to-one correspondence between types of said calls in Java format and corresponding COM methods.

45.(New) A system as set forth in claim 39 further comprising means for transforming a format of a return value from said corresponding COM object from the format of said corresponding COM object to said Java format.

46.(New) A system as set forth in claim 39 wherein said library which is compiled at compile time also includes a name of said each COM method.

47.(New) A computer program product for processing Java format calls, said computer program product comprising:

a computer readable medium;

first program instructions to compile a library which includes for each of a multiplitiy of COM methods, a return value in a COM format for said each COM method, a name of a type of COM object to contain said each COM method and an identifier for said each COM method;

second program instructions to compile from said library a Java interface which includes for each of said COM methods, a return value in a Java format for said each COM method and a name for said each COM method;

third program instructions, operable at run time and responsive to respective requests in Java format from an application to create respective proxy objects for corresponding COM objects, to generate said respective proxy objects, each of said proxy objects for the corresponding COM object programmed to:

receive a respective call in Java format from the application for the corresponding COM object, and in response, query said library for the identifier for said corresponding COM method;

transform said respective call in Java format to a call in COM format and invoke said corresponding COM method with said call in COM format; and

return to said application said return value in said Java format; and wherein

said first, second and third program instructions are stored on said medium.

48.(New) A computer program product as set forth in claim 47 wherein:

said library which is compiled at compile time also includes in COM format for each of said COM methods, types of arguments for a call to said each COM method; and

said Java interface which is complied at compile time, also includes in Java format for each of said COM methods, types of arguments for the call to said each COM method

49.(New) A computer program product as set forth in claim 48 wherein each of said calls in Java format from an application includes data defining arguments; and further comprising:

fourth program instructions using said Java interface at compile time to check validity of types of said arguments in said each of said calls in said Java format; and wherein

said fourth program instructions are stored on said medium.

50.(New) A computer program product as set forth in claim 48 wherein said identifier for said each COM method, said return value for said each COM method, said types of arguments for a call to said each COM Method, and said COM object containing said each COM method form part or all of a signature for said each COM Method.

51.(New) A computer program product as set forth in claim 48 further comprising:

fourth program instructions, responsive to said respective call in Java format from the application for the corresponding COM object, to identify from said respective call, data defining arguments of said respective call; and wherein

the call transformed into the format of said corresponding COM method includes data defining arguments corresponding to the data defining arguments of said call in Java format; and wherein

said fourth program instructions are stored on said medium.

52.(New) A computer program product as set forth in claim 47 wherein there is a one-to-one correspondence between types of said calls in Java format and corresponding COM methods.

53.(New) A computer program product as set forth in claim 47 further comprising fourth program instructions to transform a format of a return value from said corresponding COM object from the format of said corresponding COM object to said Java format; and wherein

said fourth program instructions are stored on said medium.

54.(New) A computer program product as set forth in claim 47 wherein said library which is compiled at compile time also includes a name of said each COM method.--